DOCUMENT RESUME

ED 082 470 EM 011 429

AUTHOR	Gerald, Curtis F.
TITLE	Interactive Computing With a Programmable Calculator; Student Experimentations in Numerical Methods.
INSTITUTION	California Polytechnic State Univ., San Luis Obispo.
PUB DATE	Jun 73
NOTE	8p.; Paper presented at the Conference on Computers in the Undergraduate Curricula (Claremont, California, June 18-20, 1973)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Calculation; *College Mathematics; *Electromechanical Aids; Higher Education; Interaction; *Mathematics Instruction; Program Descriptions
IDENTIFIERS	California Polytechnic University; Compucorp Model 025 Educator; Programable Calculator; San Luis Obispo

ABSTRACT

Programable desk calculators can provide students with personal experience in the use of numerical methods. Courses at California Polytechnic State University at San Luis Obispo use the Compucorp Model 025 Educator Experiences with it as a teaching device for solving non-linear equations and differential equations show that students can by-pass tedious hand calculations and elminate time-consuming program writing which other modes require; it also allows students to perform mathematical experiments analogous to the laboratory experiments of the physical sciences and to compare methods of solving the same problem. Student reactions to this mode of learning are favorable, and examination results indicate they learned more and acquired deeper understanding of those topics involving the calculator's use. Other advantages of programable calculators include the following: 1) their cost is relatively low; 2) they eliminate programing, keypunch, and syntax errors; 3) they are more available to students than the generally overloaded computer system; 4) they facilitate personal involvement by the student, perhaps due to their small size and relative simplicity; 5) their relatively slow speed promotes student awareness of complex calculations; and 6) they expose the student to a valuable educational tool. (PB)

INTERACTIVE COMPUTING WITH A PROGRAMMABLE CALCULATOR - STUDENT EXPERIMENTATIONS IN NUMERICAL METHODS

Curtis F. Gerald
Computer Science and Statistics Departments
California Polytechnic State University
San Luis Obispo, California 93401
(805) 546-2824

In a course that exposes students to the application of numerical methods, it is important that they get as much personal experience with the various methods as possible and that they be able to directly compare the relative efficiencies of these methods in solving a variety of problems. In our course in Numerical Methods, part of a year's sequence in numerical procedures and numerical analysis, we have found that programmable desk calculators have special advantages in providing such experiences. This paper discusses how these were used and assesses the results.

Any course in the applications of numerical methods must rely heavily on computational aids, for the subject attacks sophisticated mathematical problems by repetitive use of essentially arithmetic procedures. Mechanical calculation facilities are essential because doing large amounts of arithmetic by hand is time consuming, expensive, error-prone, and nauseating to students. Since computers only have the capacity to do simple arithmetic and comparison operations, their use to solve scientific problems normally requires that the algorithm built into the program be one of the type classed as numerical methods. Thus the dependence between discipline and the tool is two-way. Our students have access to a moderate size computer (IBM 360, model 40) operated in batch mode, time-sharing terminals supported by a remote computer system, and non-programmable desk calculators, but we find that programmable calculators are a meaningful addition to this spectrum of computing facilities.

The Compucorp 025 Educator

The particular programmable calculator we used was the Compucorp Model 025 Educator. It prints output on a paper tape and has the usual arithmetic capabilities. A keystroke calls microprograms to compute a variety of functions including trigonometric, hyperbolic, logarithmic, exponential, and inverse trigonometric and hyperbolic functions. Ten storage registers holding fourteen-digit numbers plus two-digit exponents can be broken into 20 half-size registers. A variety of test and branch instructions permit looping and iterative procedures to be employed, and subroutines which can be nested up to six levels can be programmed. Programming can be either from the keyboard or through punched cards. The cards can be conveniently prepared by hand from perforated stock using a Port-A-Punch. Use of the cards affords not only convenient storage of programs but access to some functions and machine operations not available from the keyboard, including bit manipulation instructions. This programmable calculator is representative of an increasing assortment of models of considerable sophistication and relatively low cost ($1500 to $3000).

Programmable calculators that have capabilities to permit student interaction are marketed by many companies, including Olivetti-Underwood, Wang Laboratories, Hewlett-Packard, Tektronix, and Monroe, as well as Computer Design Corporation, the makers of our Compucorp Educators.

Programmable calculators as a teaching device in college instruction have been discussed at earlier conferences on Computing in the Undergraduate Curricula [1-5]. These papers covered their applications in the fields of biological science, physical chemistry, differential equations, and statistics. While some simple numerical procedures were employed in these applications, most of the studies used the programmed calculator only as an equation solver and did not stress the interactive aspects.

The Numerical Methods Course

The Numerical Methods course in which programmable calculators were employed covers solution of non-linear equations, polynomial interpolation, numerical integration and differentiation, solution of ordinary differential equations, and solution of sets of linear equations. While the programmable calculator can be applied in most of these areas, we discuss here only our experience with it as a teaching device for solving non-linear equations and for solving ordinary differential equations. The text used makes reference only to batch-mode programs for computer application [6].

Students were instructed in the use of the calculator and keyboard programming during the lecture-demonstration period. This was adequate since most students had used electronic calculators previously and a student monitor was available in the calculator laboratory for assistance. A two-page summary that describes the operation of the Compucorp machine was distributed. This included an example of loading a program from cards and loading a subroutine from the keyboard that was called by the card program. The normal mode of student use of the programmed calculator involved only the use of a prepunched card program and a student-entered (from keyboard) subroutine.

In addition to this group instruction and individual assistance by the student monitor, complete instructions were provided with each set of prepunched cards that told how the cards should be loaded, the steps needed to enter initial values and parameters of the numerical method, how to initiate program execution, and how the output should be interpreted. The actual program was also listed and documented. In the process of using such "canned programs," several students learned how to program on their own, using the manufacturer's manuals [7],[8], and were able to use the calculator more sophisticatedly in this and other courses. It is important to realize that such a degree of knowledge was not essential to achieve the objectives of the course.

Typical Calculator Assignments

A typical set of materials for a program that finds a root of a non-linear equation (using Newton's method) is shown in Figures 1 and 2. The codes in Figure 2 represent octal digits that were punched into appropriate columns of the cards which were supplied to the student. Figures 3 and 4 show the materials supplied for another typical application, the solution of a first order differential equation by the Runge-Kutta fourth order method.

In addition, other programs in these application areas for numerical methods were:

1. For solving non-linear equations

    a. Method of Interval Halving
    b. Linear Interpolation
    c. Secant Method
    d. Iteration

2. For solving differential equations

    a. Simple Euler Method
    b. Modified Euler Method
    c. Milne's Method
    d. Adams-Moulton Method

The students were given assignments that made them use some judgment in exercising the programs on the calculator. The following questions were typical.

1. Find where the two curves, $y_1 = x^3 - x + 1$ and $y_2 = 2e^{-x} \cos x$, intersect, using Newton's Method. Using the Compucorp program, experiment with different starting values to find the smallest value that converges to the <u>positive</u> root and reconcile this with the graph of the function $y_1 - y_2$. Plot the errors of the successive iterates obtained during your last successful experiment and show how this provides a demonstration that Newton's method converges quadratically.

2. Find a rearrangement of the function $y_1 - y_2 = 0$, where y_1 and y_2 are defined in question 1, so that convergence to the positive root is secured. Utilizing the Compucorp program, find the range of starting values that give convergence. Compare this to the range suggested by the analytical criterion. Plot the errors of successive iterates to observe "linear convergence." Does your rearrangement converge to all the roots of the equation? Find rearrangements that do converge to each root. Compare your convergent forms with that of one of your classmates as to rates of convergence.

3. Solve for $y(1)$ given that $dy/dt = y^2e^{-t} - t$, $y(0) = 0.5$. Use the Runge-Kutta fourth order method with different step sizes (employing the Compucorp program) until you know the solution is accurate to at least six decimals. (How do you know this?) For several values of $\Delta t = h$, plot the error against step size on semi-logarithmic

paper and from this deduce the order of error for R-K on this problem. What maximum step size can be used and still retain four decimals of accuracy?

Several comments are in order. (a) Without a programmable calculator, such homework assignments would be so time-consuming as to completely frustrate the students. (b) Writing programs for either batch or time-sharing modes of computer access would also require inordinate amounts of effort. (c) The use of the programmable calculator permits assigning problems in which mathematical experimentation is performed analogous to laboratory experiments in physical sciences. (d) Comparisons of several methods on the same problem are readily made, and the results are quantitative.

Results and Conclusions

The reactions of the students were highly favorable to this mode of learning. They expressed the opinion that their interest was aroused by being personally involved in setting the parameters for the programs, and most were stimulated by the challenge of having to make personal judgements. As measured by their esponses on examinations, the students learned more and had a deeper understanding of those topics where the assignments employed programmable calculators.

The general conclusion is that student experimentation in the use of numerical methods is a highly effective teaching procedure. The author plans to extend the concept to other topics in the sequence of numerical procedure courses. The power of programmable calculators is such that this teaching method can be used in many other courses, however. It is obvious that student assignments in statistics, pre-calculus mathematics, calculus, and differential equations can be posed to employ this concept of experimentation by the student. Somewhat less obvious is the application to the teaching of physics, chemistry, and the life sciences.

Comparisons With Computers

Programmable calculators are not the only way that students can solve problems using numerical procedures wherein they must vary parameters and have the opportunity to compare the results. Batch computing, with programs either entirely student written or employing pre-written subroutines, is now a conventional method. Interactive terminals, usually implemented through a time-sharing system, is a growing means of computer access which more closely resembles our experience with programmable calculators. Programmable calculators have some special advantages over these alternatives, however.

Some of the advantages of programmable calculators are:

1. Cost. For about the same or less cost than the input device itself (keypunch or terminal) the student has access to the entire computing system. We found that two programmable calculators were entirely adequate for a class of over thirty students and the calculators were also used by others at the same time.

2. Programming ease. No computer language problems exist as the programming of subroutines was done entirely by keystrokes essentially the same as a pure desk calculator computation. Keypunch and syntax errors just don't exist.

3. Availability. There was much less competition with other students for access to the facility. With a badly overloaded computer system, as we currently have, this is an important item. In institutions with less of a critical shortage, programmable calculators can usually be located more conveniently for the student.

4. Involvement. There is a much greater sense of personal concern, partly because the size of the machine is not overwhelming and all of the facility is readily apparent. The operator is a more significant part of man-machine loop and it is easier to identify those parts of the computation which require personal judgement.

5. Awareness. One of the most important advantages actually comes from the slow computing speeds of the calculator. The student is aware of the differences in more computationally complex functions, or in procedures that require more computations per step, because he must wait for the machine to perform them. The trade-offs made in numerical analysis to achieve maximum overall efficiency are now something that the student personally experiences. In computers, computing speeds are completely below the time frame of human comprehension.

6. Knowledge. Not incidental is the student's exposure to a valuable tool; he leaves the university with a better knowledge of another aspect of the array of computing devices. In many applications, a programmable calculator is the most cost-effective computing tool and should be used instead of more expensive ones.

There are some disadvantages, of course. The lack of alphameric labeling of output makes the printed tape harder to interpret. (Some of the more expensive models of programmable calculators have alphameric output or can output through a typewriter). The rigid and limited format for output is restricting. The relatively few storage registers on our particular model makes large problems or matrix manipulations (other than for small systems, up to 4x4) beyond its capability. (Again, more sophisticated models can manipulate data arrays of as many as 512 elements). Certainly a programmable calculator would not be used for sophisticated applications in complex situations, but we have found them well adapted to the majority of problems that are useful in college level instruction and suspect they are applicable to a large number of the progessional applications of numerical procedures.

REFERENCES

1. J. D. Spain, The use of the Olivetti-Underwood Programma 101 for the mathematical modeling of biological processes. Proceedings of the Conference on Computers in the Undergraduate Curricula, 1970, 6.1-6.7.

2. F. W. Getzen, The use of small, desk-top digital computers for undergraduate physical chemistry laboratory calculations. Proceedings of the Conference on Computers in the Undergraduate Curricula, 1970, 7.64-7.69.

3. G. A. Kemper, Computer (programmable calculator) supplemented ordinary differential equations. Second Conference on Computers in the Undergraduate Curricula, 1971, 27-39.

4. J. C. Tan, Applications of a Wang calculating/computing system in the undergaraduate genetics and biometry laboratory instructions. Proceedings of the Second Conference on Computers in the Undergraduate Curricula, 1971, 181-186

5. C. I. Guiliams, Through multiple regression and three-way ANOV in sophomore level applied statistics for the behavioral and natural sciences. Proceedings of the Conference on Computers in the Undergraduate Curricula, 1972, 539-545.

6. C. F. Gerald, Applied Numerical Analysis. Reading, Mass.: Addison Wesley, 1970.

7. Operating and Programming the 025 Educator, 1971, Computer Design Corp., 12401 W. Olympic Blvd., Los Angeles, California 90064.

8. Advanced Programming Reference Manual, 1971, Computer Design Corp., 12401 W. Olympic Blvd., Los Angeles, California 90064.

Method:

1. Input, store and print x_1. Calculate $F(x_1)$, print and store.
2. Test if $F(x) = 0$.
   a. If it is, stop.
   b. If not zero, proceed.
3. Calculate $F'(x)$.
4. Test if $F'(x) = 0$.
   a. If it is, print IDEN = -1 and stop.
   b. If not zero, proceed.
5. Compute $x = x_1 - F(x_1)/F'(x_1)$. Replace x_1 with x and loop back to step 1.

Directions for use of programs:

Loading the program and subroutines:

1. Press RESET, TO ( ), 0, feed the card to reader.
2. Load first subroutine, to calculate $F(x)$, from keyboard. Press TO ( ), 2, latch LOAD. Program the subroutine, avoiding registers 1 and 2. The value of x_1 is in register 1. Unlatch LOAD.
3. Load second subroutine, to calculate $F'(x)$, from keyboard. Press TO ( ), 6, latch LOAD. Program the subroutine, avoiding registers 1 and 2. The value of x is in register 1. Unlatch LOAD.

Operation:

1. Press TO ( ), 0, RESUME.
2. Enter x_1 , press RESUME. The value of x_1 and $F(x_1)$ will print, and successive values of x and $F(x)$ until $F(x) = 0$. If $F'(x) = 0$, the number -1 will print as an identifier, a number stripped of its decimal portion.

Figure 1. Compucorp Program for Newton's Method.

Compucorp®

LOAD AT (0)
LOAD SUBR. FOR F(X) AT (2)
LOAD SUBR. FOR F'(X) AT (6)

PROGRAM NEWTON'S METHOD

PAGE 1 OF 1 DATE 9/25/72 NO. 1-3

BR PT	STEP	INSTR.	CODE	REMARKS
40	00	HALT	401	
	01	STR 1	440	Enter x_i, store, print
	02	PRTX	026	
	03	TO(2)	742	
	04	STR 2	441	Compute f(x), store, print
	05	PRTA	027	
	05	IENO	437	Is f(x) ≠ 0?
	07	THEN	540	
	10	JUMP	612	Yes, proceed.
	11	HALT	401	f(x) = 0, so stop.
	12	TO(6)	746	Calculate f'(x).
	13	IENO	437	Is f'(x) ≠ 0?
	14	THEN	540	
	15	JUMP	623	Yes, proceed.
	16	1	001	
	17	CHSE	013	
	20	IDEN	732	f'(x) = 0. Print IDEN = -1 and stop.
	21	PRTX	026	
	22	HALT	401	
	23	1/x	054	
	24	×	070	
	25	RCR 2	461	
	26	CHSE	013	Calculate new x. Replace x_i
	27	+	060	with new x.
	30	RCR 1	460	
	31	=	020	
	32	STR 1	440	
	33	PRTX	026	Print next x and repeat.
	34	JUMP	603	
	5			
	6			
	7			

BR PT	STEP	INSTR.	CODE	REMARKS
	0			
	1			
	2			
	3			
	4			
	5			
	6			
	7			
	0			
	1			
	2			
	3			
	4			
	5			
	6			
	7			
	0			
	1			
	2			
	3			
	4			
	5			
	6			
	7			
	0			
	1			
	2			
	3			
	4			
	5			
	6			
	7			

9400037 (51)

Figure 2

Method:

x_0 is in register 1, y_0 in registers 2 and 3 by hand to save program steps (allowing only two cards)

1. Input h and store the value of h/2.

2. Calculate $f(x,y) = dy/dx$ by subroutine. Store.

3. Recall values of x, y, and f and print.

4. Increment y_0 by $1/2k_1 = 1/2h*f(x,y)$ (in register 3).

5. Increment x by 1/2h.

6. Increment y_0 by $1/2k_2 = 1/2h*f(x+1/2h,y+1/2k_1)$ (in register 3).

7. Increment y_0 by $k_3 =h*f(x+1/2h,y+1/2k_2)$ (in register 3).

8. Increment x by 1/2h again, now equals x_1.

9. Calculate $y_1 = y_0 +(h/6)*(k_1 +2k_2 + 2k_3 +k_4)$.

10. Put y_1 in registers 2 and 3 and loop back to (3).

Directions for use of program:

Loading the program and subroutine:

1. Press RESET, TO ( ), 0, feed cards to reader.

2. Press TO ( ), 4, latch LOAD, then enter the subroutine from the keyboard that calculates $f(x,y) = dy/dx$. Unlatch LOAD. (Get x from register 1, y from register 3.)

Operation:

1. Put x_0 into register 1, y_0 into registers 2 and 3.

2. Press TO ( ), 0, RESUME.

3. Enter h, press RESUME. The successive values of x, y, and f(x,y) print, with x values differing by h. When sufficient values have been printed, terminate by RESET, CLEAR X, or STOP.

Figure 3. Compucorp Program for Runge-Kutta Method.

Compucorp®

LOAD AT (0)
LOAD SUBR. FOR $dy/dx = f(x, y)$ AT (4)

PROGRAM RUNGE-KUTTA METHOD

PAGE 1 OF 1 DATE 11/1/72 NO. 6.3

BR PT	STEP	INSTR	CODE	REMARKS
40	0	HALT	401	Enter h
	1	÷	072	
	2	2	002	Store $h/2$
	3	=	020	
	4	STR4	443	
	5	TO(4)	744	Calculate $f(x,y)$, store
	6	STR5	444	
	7	RCR1	460	
	10	PRTX	026	
	1	RCR2	461	Recall x, y, f and print
	2	PRTX	026	
	3	RCR5	464	
	4	PRTA	027	
	5	X	070	
	6	RCR4	463	
	7	+	060	Increment y_0 in reg 3 by $1/2\ k_1$
	20	RCR2	461	
	1	=	020	
	2	STR3	442	
	3	RCR4	463	
	4	+n	035	Increment x_0 by $1/2\ h$
	5	1	001	
	6	TO(4)	744	
	7	+n	035	
	30	5	005	Calculate $f_{1/2}$, add $2f_{1/2}$ to f_0
	1	+n	035	
	2	5	005	
	3	X	070	
	4	RCR4	463	
	5	+	060	Increment y_0 by $1/2\ k_2$, store in reg 3
	6	RCR2	461	
	7	=	020	

BR PT	STEP	INSTR.	CODE	REMARKS
	40	STR3	442	
	1	TO(4)	744	
	2	+n	035	
	3	5	005	Calculate $f_{1/2}'$, Add $2f_{1/2}'$ to sum.
	4	+n	035	
	5	5	005	
	6	X	070	
	7	RCR4	463	
	50	X	070	
	1	2	002	Increment y_0 by k_3, store in reg 3
	2	+	060	
	3	RCR2	461	
	4	=	020	
	5	STR3	442	
	6	RCR4	463	
	7	+n	035	Increment x by $1/2\ h$ again
	60	1	001	
	1	TO(4)	744	
	2	+n	035	Calculate $f(x_0+h, y_0+k_3)$, add to sum.
	3	5	005	
	4	RCR5	464	
	5	X	070	
	6	RCR4	463	
	7	÷	072	
	70	3	003	Compute new y, $y = y_0 + \frac{1}{6}(k_1+2k_2+2k_3+k_4)$, store in regs 2 and 3.
	1	+	060	
	2	RCR2	461	
	3	=	020	
	4	STR2	441	
	5	STR3	442	
	6	JUMP	605	Loop back for next interval
	7			

9400037 (51)

Figure 4